



Minimum Cost Palindromic Deletions

Problem Statement

Oski has received a string S of length N consisting of lowercase English letters as a gift. However, Oski only likes strings that are palindromic—that is, strings that read the same backward as forward.

To transform S into a palindrome, Oski can delete a (possibly empty) subset of characters from S . Each character at 1-based index i has an associated deletion cost given by $C[i]$.

The total cost of deleting a set of characters is the sum of their individual deletion costs in C . Oski wants to find the **minimum total cost** required to make the remaining sequence of characters form a palindrome.

Note: An empty string is considered a valid palindrome with a length of 0.

Input Format

- The first line contains a single integer T ($1 \leq T \leq 100$) — the number of test cases.
- For each testcase:
 - The first line contains an integer N ($1 \leq N \leq 1000$) — the length of the string S .
 - The second line contains the string S of length N .
 - The third line contains N space-separated integers $C[1], C[2], \dots, C[N]$ ($1 \leq C[i] \leq 10^9$) — the cost of deleting each character.

The sum of N over all test cases does not exceed 20,000.

Output Format

For each test case, print a single integer — the minimum total cost of deletions required to make the remaining string a palindrome.

Sample Input

```
3
7
abacaba
1 2 3 4 5 6 7
5
radar
10 20 30 20 10
4
abcd
5 1 1 5
```

Sample Output

0
0
7

Explanation

Test Case 1

The string `abacaba` is already a palindrome, so Oski doesn't need to delete any characters. The total cost is 0.

Test Case 2

The string `radar` is already a palindrome. The total cost is 0.

Test Case 3

For the string `abcd` with costs $[5, 1, 1, 5]$:

- To leave `a` (delete `b`, `c`, `d`), the cost is $1 + 1 + 5 = 7$.
- To leave `d` (delete `a`, `b`, `c`), the cost is $5 + 1 + 1 = 7$.
- To leave an empty string (delete all), the cost is $5 + 1 + 1 + 5 = 12$.

The minimum total cost to leave a valid palindrome is 7.